

# Sensor Task Manager (STM)

Timothy J. Rogers  
V.S. Subrahmanian  
University of Maryland

# Talk Outline

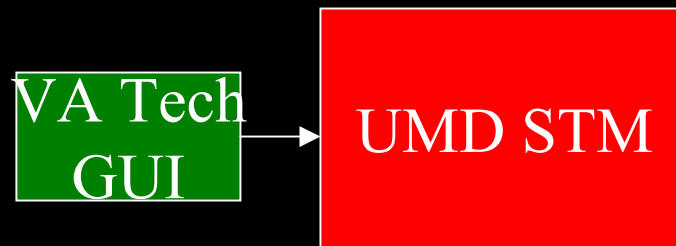
- Architecture of Tasking Component
- Probabilistic databases
- STM Task Merger Component
- STM Multiview Action Module
- STM Heterogeneous Data/Software Access components
- Ongoing/future work

# Tasking Architecture

VA Tech  
GUI

Where to monitor?

# Tasking Architecture



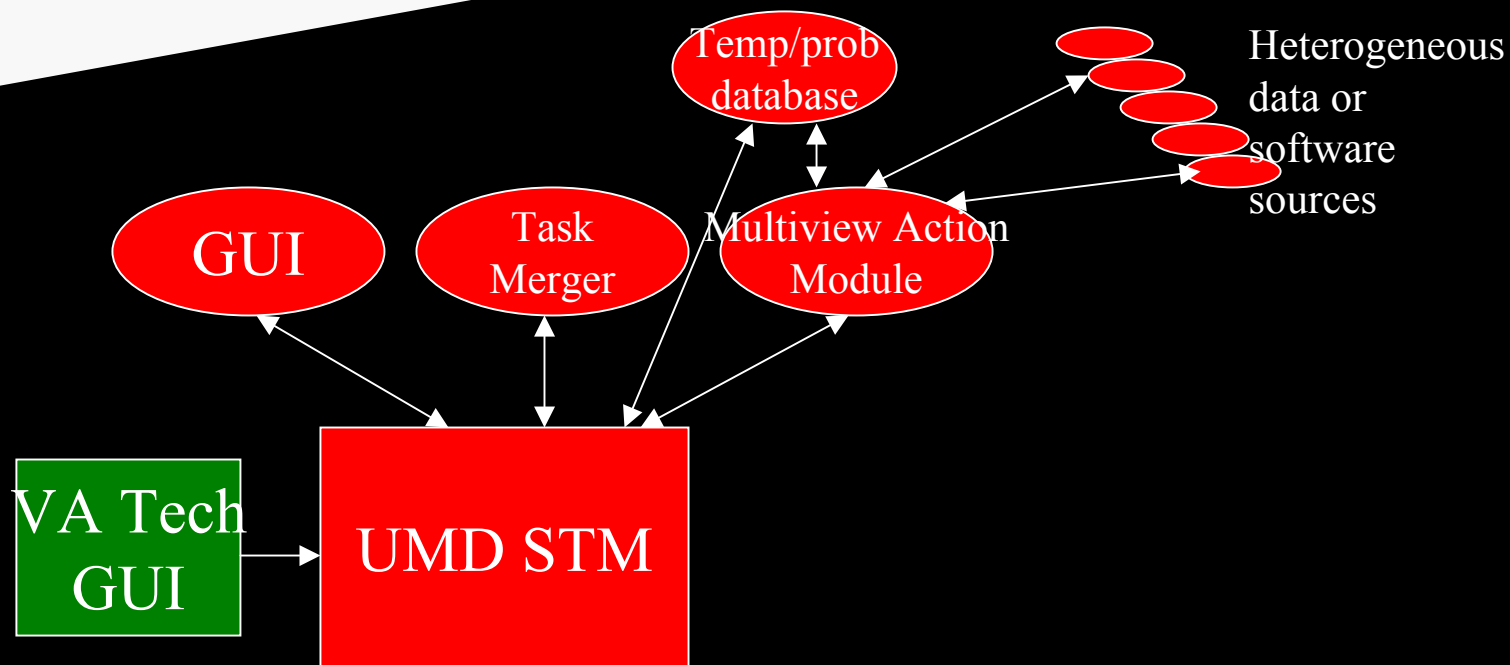
What to monitor?

When to monitor?

What sensed conditions to look for?

What to do if those conditions hold?

# Tasking Architecture



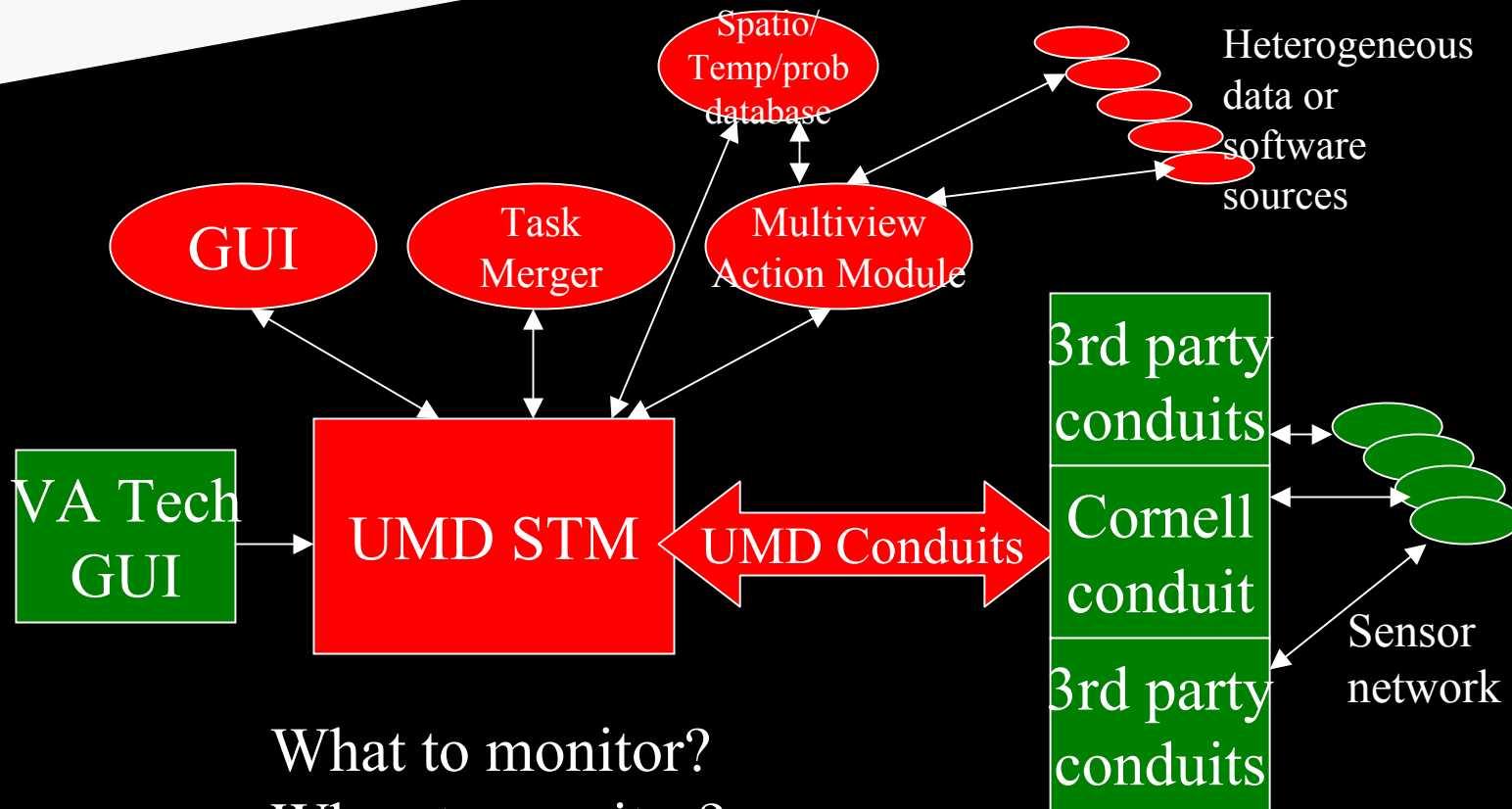
What to monitor?

When to monitor?

What sensed conditions to look for?

What to do if those conditions hold?

# Tasking Architecture



What to monitor?

When to monitor?

What sensed conditions to look for?

What to do if those conditions hold?

# Temporal/Probabilistic (TP) Database

- Sensory data is spatio-temporal (when/where was the event sensed) ?
- It is also uncertain.
- STP-databases store uncertain data with temporal components.
- We've made a first cut effort to build models involving time and uncertainty.
- Built on top of ODBC.

## KEY CONTRIBUTIONS

- TP-relational algebra
- Soundness/completeness results.
- Complexity results
- Query optimization (ongoing)
- Implementation and experiments.

ACM Trans. On Database Systems, 2001

# STM Task Merger

- Takes as input a set of tasks.
- A task is expressed in a high-level tasking language.
- This gets broken down into subtasks.
- Different tasks may share common subtasks.
- Even syntactically different subtasks may be "equivalent" or mergeable.
- Merging tasks reduces load.



# STM Task Merger Example

- A highway has eight exits (of interest).
- Task A: Monitor exits A,B,C for all .
- Task B: Monitor exits C,D for TELs.
- Tasks not equivalent. Yet, can be merged to reduce load on sensor network.
- One solution: Monitor exits A-D. Identify vehicles offline. Extract answers to tasks.
- Many other solutions possible.
- Pick solution that optimizes some objective.

# STM Task Merging Contributions

- When are two tasks equivalent?
  - NP-complete under some conditions
  - Undecidable otherwise
  - Devised conditions (sufficient) which are computable and guarantee equivalence, but not conversely (of course).
- Sound/complete algorithms for checking these sufficient conditions.
- Techniques to cost tasks.
- Techniques to merge a set of tasks into an expected minimal cost set.
  - Provably optimal algorithms
  - Greedy and other approximations.
  - Experiments
- Techniques to “Optimally” (w.r.t. an objective function) split a set of tasks into manageable subsets.

IJCAI-2001 paper. Another paper submitted to ACM Trans. On CL

# STM Multiview Action Module

- Different users may express different conditions that they care about.
- User A: if the total number of TELs seen exiting exits C,D exceeds 2 per day, then analyze images and classify vehicles, chart vehicle types via a graphic, and send graphic to all people monitoring the region.
- User B: if the total number of TELs seen exiting exits C,D exceeds 2 per day and thermal sensors indicate condition  $\alpha$ , then do something else.

# STM Multiview Action Module

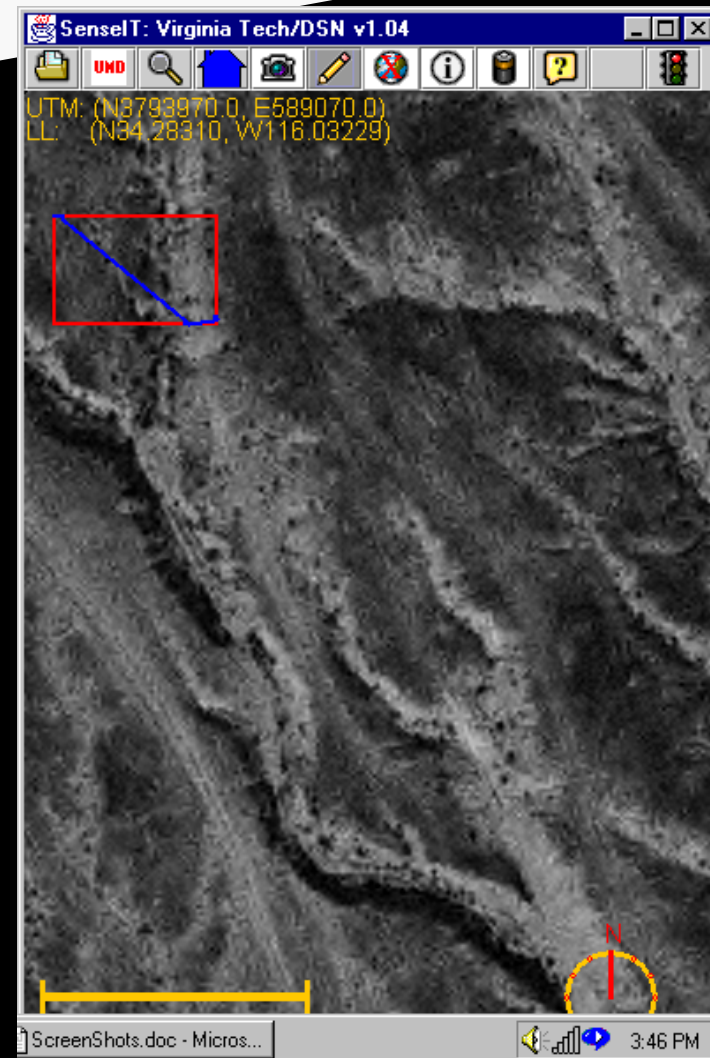
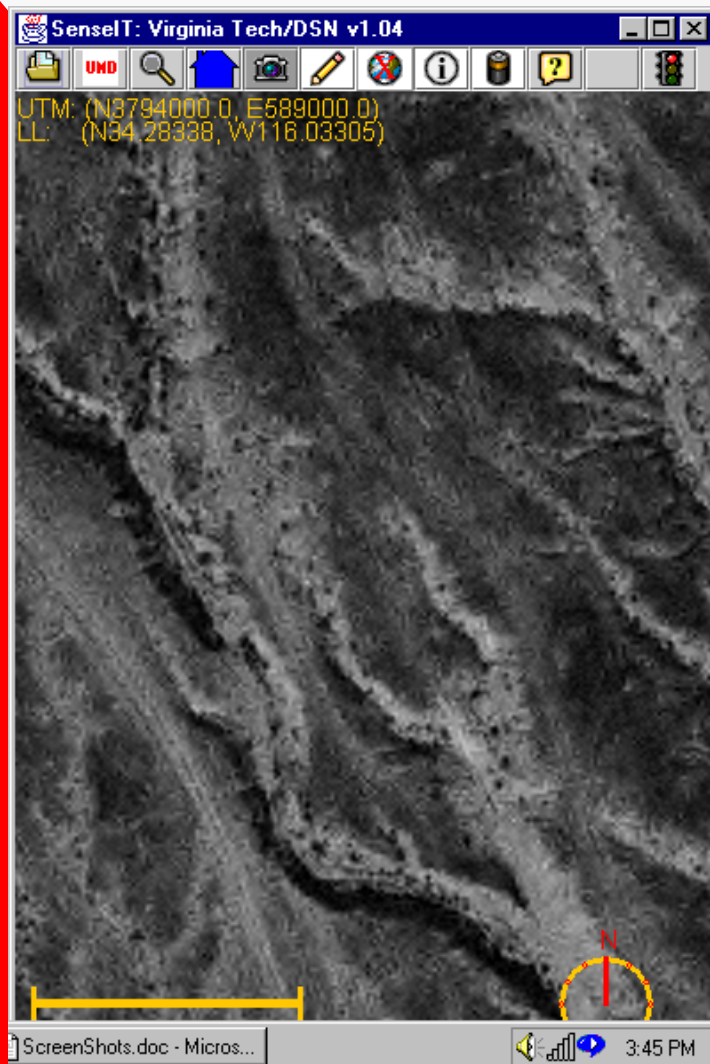
- Both examples are of the form:
  - if condition holds then do something
- The “do something” part may also have a temporal component (e.g. now, within T time units, etc.)
- Hence, lots of conditions need to be monitored.
- Conditions span multiple data sources and analytic tools
- Newly sensed data may affect conditions and may necessitate actions.
- Merge conditions to reduce load. This is particularly important as sensory changes occur frequently and evaluating conditions sequentially involves unacceptable redundant computations.

Work on temporal actions in AI Journal, 2001

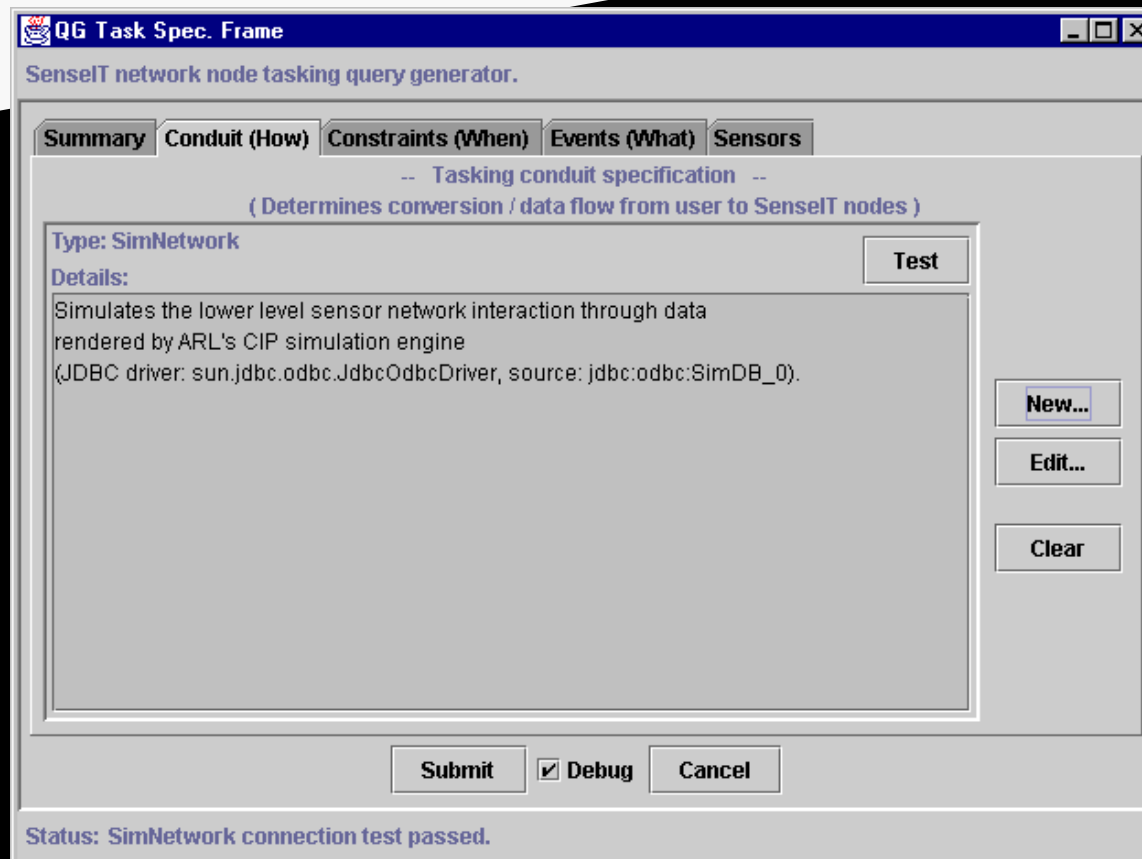
# Ongoing/Future Work

- Spatial component to build spatio-temporal object bases (STOB).
- Work on optimizing STOB-database queries.
- Algorithms to merge multiviews over heterogeneous data source calls and software source calls.

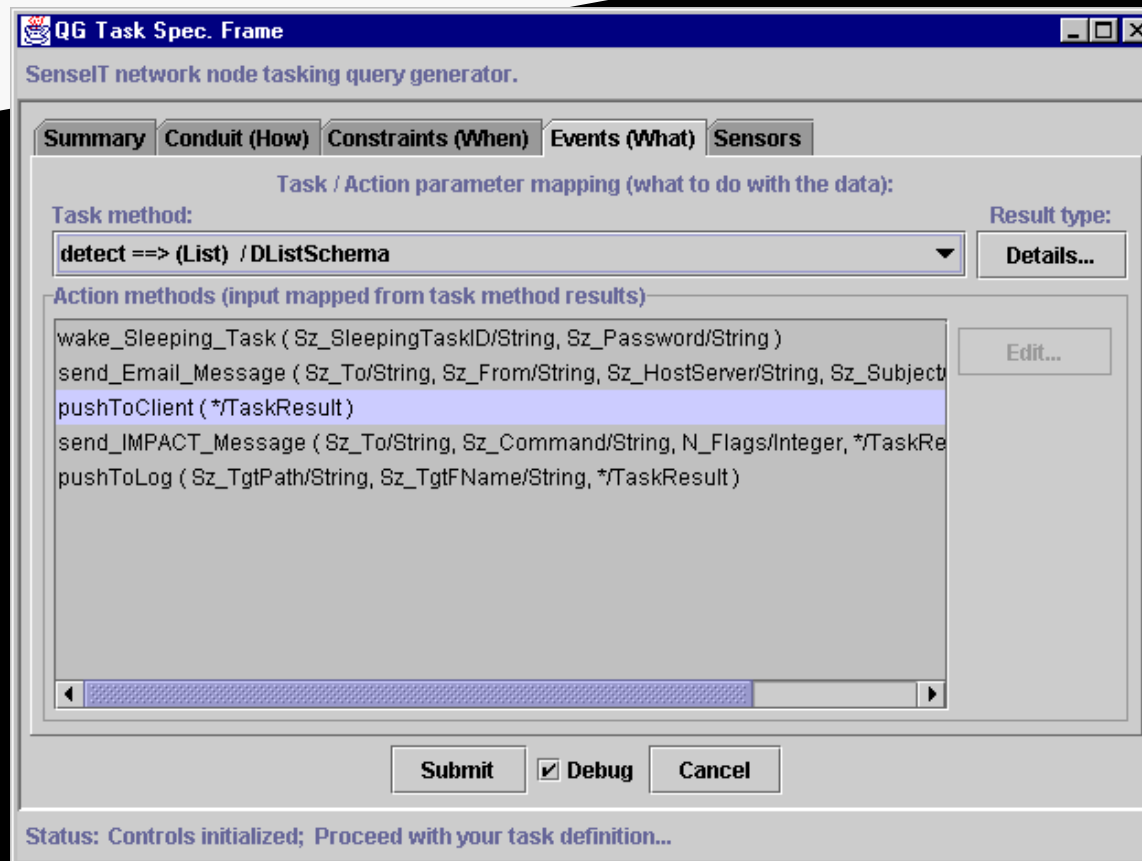
# Screenendumps



# Screenendumps

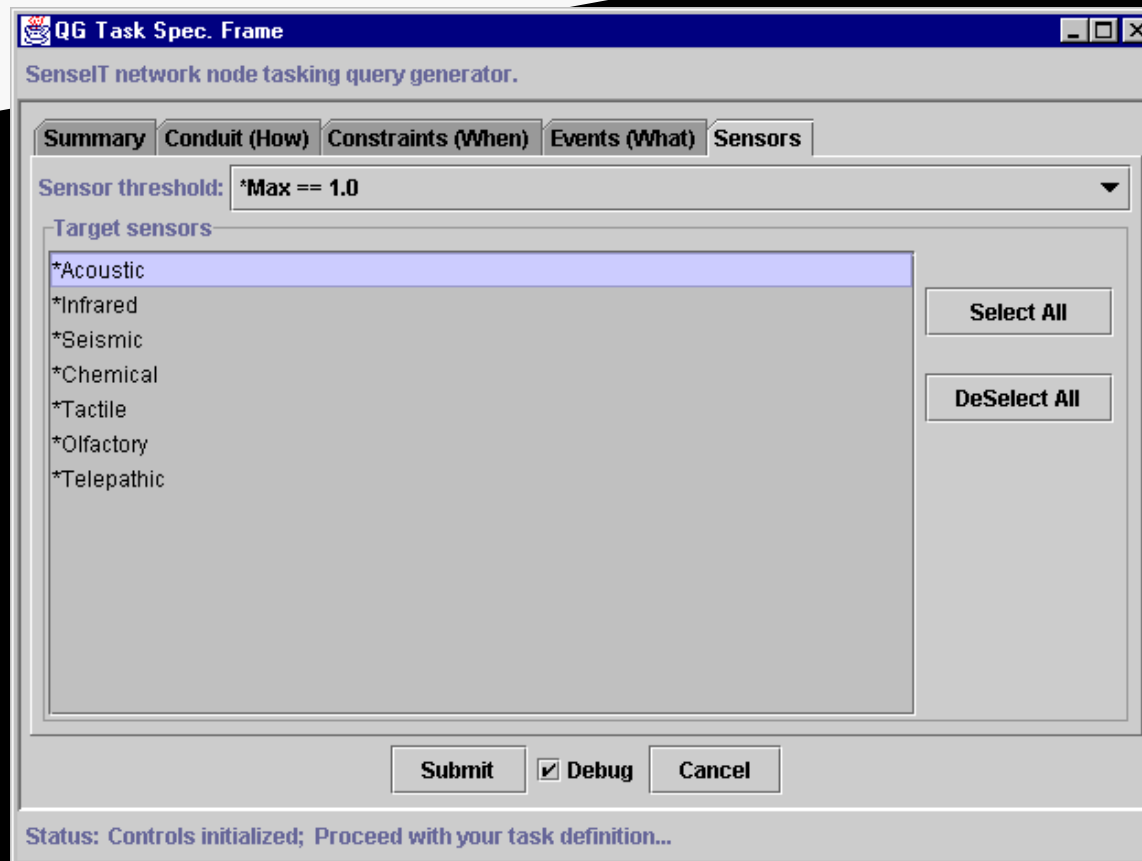


# Screenendumps





# Screenendumps



# Contact Info

- V.S. Subrahmanian
- Dept. of Computer Science  
AV Williams Building  
University of Maryland  
College Park, MD 20742.
- Tel: (301) 405-2711
- Fax: (301) 405-8488
- Email: [vs@cs.umd.edu](mailto:vs@cs.umd.edu)
- URL: [www.cs.umd.edu/users/vs/index.html](http://www.cs.umd.edu/users/vs/index.html)